REMARKS

This amendment is submitted in response to the Office action mailed 7/07/03. Claims in their pending forms are also submitted herein.

I. Claims 10-13, 15-17, 21-24, and 26-28 were rejected in the Office actions mailed January 27, 2003 and July 7, 2003, based on 35 U.S.C. 102(e), citing U.S. Patent 6,446,142 to Shima et al. (referred to herein as "Shima.") **Although Applicant responded to these rejections in an amendment dated 3/27/03, Applicant's arguments were not addressed in the Advisory Action dated 4/8/03. Applicant once again submits the following arguments against Examiner's rejections based on Shima and 35 USC 102(e).**

Before analyzing Shima, it is useful to review claim 10 of the present application. Claim 10 reads:

> A method for providing a transaction layer for a module having at least one node connected to a serial bus that configures a link device for each of said at least one nodes comprising:
>
> detecting a link driver;
>
> receiving capabilities of said link driver;
>
> generating a link driver configuration for said link driver from said capabilities of said driver; and
>
> loading said link driver configuration into said link driver.

A. Shima does not teach disclose or otherwise suggest detecting a link driver as claimed in claims 10 and 21.

Link layers and link drivers are well known by those skilled in the art. Shima also discusses the link layer at column 1, lines 54-56:

> Each node on the IEEE 1394-1995 bus structure has a 16 bit node ID. The node ID is the address that is used for data transmission on the data link layer.

Shima further discusses the link layer at column 2, lines 24-28:

> The link layer 14 provides data packet delivery service for both asynchronous and isochronous data packet transport. This supports both asynchronous data transport, using an acknowledgement protocol, and isochronous data transport, providing real-time guaranteed bandwidth protocol for just-in-time data delivery.

However, Shima is not concerned with detecting a link driver, receiving its capabilities, generating a link driver configuration, or loading a link driver configuration into a link driver. Rather, Shima (as stated in column 3, lines 34-37) states:

> A controlling application utilizes existing handle objects, as appropriate, to reconfigure objects to dynamically enumerate and represent devices coupled to a serial bus network after a bus reset event. Preferably, the serial bus network is an IEEE 1394-1995 serial bus network.

Shima continues at column 3, lines 46-47, to state that a handle includes a 64-bit unique identifier value that is compared to the objects to find a match. By this statement, Shima obviously is NOT talking about a link driver, for as stated above on col 1, lines 54-56, the 16-bit node ID is the address that is used for data transmission on the link layer.

Column 4, lines 1-31 in Shima states:

[In one aspect of the invention, a method of representing functions and characteristics of a device comprises the steps] of maintaining a library of a plurality of available subobjects each representing an available subunit, determining characteristics of the device, including resident subunits within the device, retrieving retrieved subobjects from the library corresponding to the resident subunits and assembling the retrieved subobjects into an object representing the functions and characteristics of the device. The method further comprises the step of receiving self identifying information for the device. The characteristics of the device are determined from the self identifying information. The device is preferably coupled within an IEEE 1394 serial bus network. The device is preferably an audio/visual device. The device is a remote device and the object is maintained by a local device.

In another aspect of the invention, an apparatus for representing functions and characteristics of a device comprises means for accumulating data about the functions and characteristics of the device and a controlling circuit coupled to the means for accumulating data for maintaining an object representing the functions and characteristics of the device, wherein the object is assembled by determining resident subunits within the device, retrieving retrieved subobjects from a library of available subunits corresponding to the resident subunits and assembling the retrieved subobjects into the object representing the functions and characteristics of the device. The resident subunits are determined from self identifying information received by the means for accumulating. The apparatus further comprises an interface circuit coupled to the means for accumulating data and configured for coupling to a network of devices for communicating with the network of devices. Preferably, the network of devices is an IEEE 1394 serial bus network.

There is no mention of a link layer or detecting a link driver in Shima in the text cited above.

Column 10, lines 48-57 in Shima states:

> Subobjects for each subunit within a device are assembled into an object representing the device. When the device reports its self-identifying information, it is determined which subunits are included within the unit. Subobjects representing these subunits are copied from a subobject library and assembled into the object representing this unit. In this manner, objects are generated for units of many different configurations, including units which do not yet exist at the time of the development of the controlling application.

Likewise, there is no discussion of detecting a link driver, or any discussion whatsoever of a link layer. **For these reasons, Shima does not teach or otherwise suggest detecting a link driver.**

B. Shima does not teach disclose or otherwise suggest receiving capabilities of the link driver as claimed in claims 10 and 21.

It is stated in the Office Action dated 1/27/03 that Shima at column 3, lines 13-17, column 3, lines 38-40, and column 4, lines 1-31 teaches receiving capabilities of said link driver.

Column 3, lines 13-17 in Shima states:

> [Generally, such a controlling] or monitoring application maintains a representation or object of each device. This object represents the capabilities of the device. This object is typically copied from a library of objects representing known devices. However, if a new device, without a representative object in the library, is connected to the network, the

> controlling or monitoring application is at a loss for representing this new device.

There is no mention in this text of a link driver, or even a link layer, much less receiving capabilities of a link driver.

In column 3, lines 38-40, Shima states:

> During a self-identifying process, after the bus reset, information about the characteristics of the devices within the network is received.

Again, there is no mention in this text of a link driver, or even a link layer, much less receiving capabilities of a link driver.

C. Shima does not teach disclose or otherwise suggest generating a link driver configuration for the link driver from the capabilities received as claimed in claims 10 and 21.

It is stated in the Office Action dated 1/27/03 that Shima teaches generating a link driver configuration for a link driver from capabilities of the driver at column 3, lines 41-42 and column 4, lines 1-31.

Column 3, lines 41-42 in Shima states:

> From this self-identifying information objects representing the devices are generated.

There is no mention here of a link driver, nor detecting the capabilities of a link driver, no generating a link driver configuration in Shima as cited above. Nor is there any discussion of this in Shima at column 3, lines 1-31, shown above. It cannot be argued that subunits as discussed in Shima are comparable to link drivers as discussed in the present application.

D. Shima does not teach disclose or otherwise suggest loading the link driver configuration in the link driver as claimed in claims 10 and 21.

It is also stated in the Office Action dated 1/27/03 that Shima at column 3, lines 33-37, and column 4, lines 1-31 teaches loading a link driver configuration into a link driver.

Column 3, lines 33-37 in Shima states:

> A controlling application utilizes existing handle objects, as appropriate, to reconfigure objects to dynamically enumerate and represent devices coupled to a serial bus network after a bus reset event.

There is no discussion here of loading a link driver configuration for a link driver from capabilities of the driver. Likewise, Shima, in column 4, lines 1-31, presented above, does not discuss loading a link driver configuration for a link driver from capabilities of the driver.

Applicant has addressed every aspect of the rejection of claim 10 under 35 U.S.C. 102(e) as cited in the Office Action dated 1/27/03. Claims 11, which depends from claim 10, incorporate the limitations of claim 10, and thus a rejection of claim is argued against as well. Claim 21, a claim that mirrors claim 10, is also asserted to be patentable by the above discussion. As claim 22 depends from claim 21, Applicant asserts that claim 22 is patentable as well, despite the cited prior art.

As to claims 12 and 23, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 13 and 24, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 15 and 26, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 16 and 27, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 17 and 28, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

II. Claims 10-13, 15-17, 21-24, and 26-28 were rejected in the Office action mailed July 7, 2003, based on 35 U.S.C. 102(e), citing U.S. Patent 6,253,255 to Hyder et al. (referred to herein as "Hyder.")

A. Hyder does not teach disclose or otherwise suggest detecting a link driver as claimed in claims 10 and 21.

In the Office action mailed 7/07/03, Examiner cites Hyder at col. 4, lines 5-9, and col. 7, lines 7-9, as disclosing the limitation of claims 10 and 21 of detecting a link driver. Hyder at col. 4, lines 5-9 reads:

> The transport layer subsequently issues a multi-packet or send packets request to the abstract interface which in turn evaluates the device driver or descriptor as specified by transport driver to determine capabilities or sophistication of the destination link layer device driver.

Examiner is again reminded of the claim language of claim 10 (similar limitations appear in claim 21):

> A method for providing a transaction layer for a module having at least one node connected to a serial bus that configures a link device for each of said at least one nodes comprising:
>
> detecting a link driver;

receiving capabilities of said link driver;

generating a link driver configuration for said link driver from said capabilities of said driver; and

loading said link driver configuration into said link driver.

Nowhere in this text does Hyder mention a link driver, much less detecting a link driver. It cannot be argued that *evaluating a device driver as specified by a transport driver* is functionally similar to *detecting a link driver.*

Hyder at col. 7, lines 7-9 reads:

> Send packets request 226 may also be comprised of a device handle or identifier when a plurality of link layer device drivers are present.

Nowhere in this text does Hyder mention detecting a link driver. It cannot be argued that a send packets request is functionally similar to detecting a link driver. Also, "when a plurality of link layer device drivers are present" of Hyder in this text is functionally dissimilar from detecting a link driver. A link driver is not a link layer device driver.

B. Hyder does not teach disclose or otherwise suggest receiving capabilities of the link driver as claimed in claims 10 and 21.

Hyder at col. 4, lines 42-44 and col. 11, lines 37-41 is cited as disclosing receiving capabilities of said link driver. Col. 4, lines 42-44 of Hyder reads:

> Sophistication of capability information of a driver is loaded into the abstract interface upon loading the driver into the system.

Here, Hyder is describing a device driver, not a link driver. See Hyder, col. 4, lines 37-38, immediately preceding the cited text: "For example, the abstract interface upon receiving a multi-packet transfer request, evaluates the sophistication and capability of the designated device driver." Hyder does not teach, suggest, or otherwise disclose

receiving capabilities of a link driver. Link drivers and device drivers are functionally dissimilar. For example, see **information block 304 of FIG. 5 of Hyder is described at col. 10, lines 60-64**: Abstract interface 220 further provides the capability for a driver to query abstract interface 200 for determining specific configurations, statistics, and capabilities of **device drivers** resident within driver interconnection/capability information block 304.

C. Hyder does not teach disclose or otherwise suggest generating a link driver configuration for the link driver from the capabilities received as claimed in claims 10 and 21.

Examiner cites Hyder at col. 8, lines 13-22 as disclosing the limitation of claim 10 and 21 of generating a link driver configuration for said link driver from said capabilities of said driver. Hyder at col. 8, lines 13-22 reads:

> Capability information of link layer device driver 324 is incorporated into abstract interface 220 upon the loading or configuration of link layer device driver 324 into the present computer system. By incorporating capability information into abstract interface 220, link layer device drivers and transport drivers having varying capabilities may interoperate due to the mediation capabilities of abstract interface 220 to accommodate or supplement the functionality lacking in less capable or sophisticated drivers.

Attention is directed specifically to the language in this cited text,

> "Capability information of link layer device driver 324 is incorporated into abstract interface 220 upon the loading or configuration of link layer device driver 324 into the present computer system."

This is functionally dissimilar from generating a link driver configuration as link drivers are not device drivers, and also because the abstract interface of Hyder is not a link driver configuration.

D. Hyder does not teach disclose or otherwise suggest loading the link driver configuration in the link driver as claimed in claims 10 and 21.

Examiner again cites Hyder at col. 8, lines 13-22 and a signal arrow from ref No. 364 in Fig. 5 of Hyder as disclosing the limitation of claims 10 and 21 of loading said ilink driver configuration into said link driver. Col. 8, lines 13-22 has been discussed above. Applicant's argument above applies to this rejection as well. Nowhere does Hyder mention a link driver. All Hyder discusses in terms of drivers are device drivers.

Claim 11, which depends from claim 10, incorporate the limitations of claim 10, and thus a rejection of claim is argued against as well. Claim 21, a claim that mirrors claim 10, is also asserted to be patentable by the above discussion. As claim 22 depends from claim 21, Applicant asserts that claim 22 is patentable as well, despite the cited prior art.

As to claims 12 and 23, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 13 and 24, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 15 and 26, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 16 and 27, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

As to claims 17 and 28, these claims incorporate the limitations of claims 10 and 21, discussed above, and thus are patentable over the cited prior art.

III. 35 USC 103(a) rejections:

Examiner has repeated the same 35 USC 103(a) rejections in the Office action of 7/07/03 as appeared in the final Office action dated 1/27/03. **Although Applicant responded to these rejections in the response dated 3/27/03, Applicant's arguments were not addressed in the Advisory Action dated 4/8/03. Applicant once again submits the following arguments against Examiner's rejections based on 35 USC 103(a).**

Claims 14, 18, 25, and 29 are rejected under 35 USC 103(a) as being unpatentable over Shima. However, as Shima does not teach the base claims for the present application, although Shima discusses a link layer in the background of the invention, and reproduced above, Shima teaches something completely different from the present invention. As there is no mention in Shima of the limitations of the base claims of the present application, as discussed in detail above, one skilled in the art would not see the present invention as obvious in light of Shima.

Claims 19, 20, 30 and 31 are rejected under 35 USC 103(a) as being unpatentable over Shima and further in view of Levy, US patent 6,212,633. In the Office Action dated 1/27/03, it is stated that Levy teaches a method for configuring a link device of aP1394 serial bus based on capabilities (column 9, lines 10-26) of a link driver and an input of user-defined configuration data received (column 10, lines 43-55).

Levy states at column 9, lines 10-26:

> A serial bus management layer 128 is also supported in node 100 to support interface configuration and management activities for the node. The precise bus management support included are dependent upon the capabilities of the node. The serial bus management layer of each node typically must include configuration functions, while other bus management functions are optional (e.g., power management). For example, functions such as cycle master, isochronous resource management and/or bus manager may be implemented in one or more of a

> plurality of nodes coupled to the communications interface. As such, layer 128 may include, for example, a bus manager component 130 and an isochronous resource manager 132. Layer 128 may also include a node controller 134 that handles various housekeeping tasks for the node, including, among other tasks, segmenting resources and supporting IEEE 1212 configuration and status registers (CSR's).

There is no mention in this citation above of anything having to do with the link layer or the link driver.

Levy, at column 10, lines 43-55 states:

> Another type of secured or selective-secured node is an interactive-type node, which has a mechanism to build a permanent authorization list by asking an external resource whether or not located devices on the interface should be authorized for communication with the node. The external mechanism may be, for example, a display panel or combination of buttons that a user is required to supply input to manually determine which devices are authorized for communication with a particular node. Other external mechanisms may be, for example, an on-screen walk-through setup menu, a configuration or properties dialog box in a computer graphical user interface (GUI), a keypad, a telephone keypad, and other suitable known user input mechanisms.

Again, there is no mention in this citation above of anything having to do with the link layer or the link driver. It cannot argued that without describing at least some of the limitations of the claims of the present application that Levy would render the present invention obvious over another cited reference that does not disclose any of the claimed elements in the present application. For this reason, Applicant respectfully submits that the 35 USC 103(a) rejections of claims 19, 20, 30 and 31 are invalid based on neither

Levy nor Shima teaching, disclosing, or otherwise suggesting the limitations of the claims of the present application.

Respectfully submitted,
Sierra Patent Group, Ltd.

Dated: October 7, 2003

Sierra Patent Group, Ltd.
PO Box 6149
Stateline, NV 89449
(775) 586-9500

John W. Crosby
Reg. No. 49,058